Amendments to the Specification:

Please replace the paragraph beginning at page 2, line 3, with the following paragraph:

Moreover, existing service networks have faced some resistance due to perceived security problems with the connection of client systems to the service provider's network which may limit the security of both networks. Accordingly, a robust service network that is dynamically configurable and secure is desirable.

Please replace the paragraph beginning at page 2, line 10, with the following paragraph:

In one aspect the present invention relates to a method for dynamically extending a firewall. The method includes the step of establishing a connection with a remote system. A connection, in some embodiments a serial connection,[[;]] is initiated with the remote system and the remote system assigns identifiers to the local system. In some embodiments, the identifier is an IP address transmitted to the client system.

Please replace the paragraph beginning at page 3, line 7, with the following paragraph:

FIGs. 5A and 5[[A]]B are screen shots depicting exemplary embodiments of user interfaces for controlling the booting process;

Please replace the paragraph beginning at page 3, line 11, with the following paragraph:

FIGs. 8, 8A, and 8B together are [[is]] a functional flow diagram ~~of one embodiment~~ of the steps to be taken to initiate a client connection from a service network in accordance with one embodiment of the invention;

Please replace the paragraph beginning at page 7, line 7, with the following paragraph:

Each I/O controller 24 includes service management logic which performs various system management functions, such as: monitoring the operational status of the system; performing on-line diagnostics of the system; and providing an interface for remotely viewing system operation (including a processor boot sequence). In some embodiments, the service management logic includes a modem providing a serial line connection to a service network. In other embodiments, the service management logic includes a connection for communicating with other customer equipment, such as an Ethernet connection [[of]] or another local area network connection. In

some embodiments, the service management logic is provided as a separate board that is in communication with I/O controller 24. In one particularly preferred embodiment, a service management board including all service management logic connects to I/O controller 24 via a PCI slot. The service management logic (referred to hereafter as SML) may be provided with a power supply separate from the remainder of the system 14'.

Please replace the paragraph beginning at page 10, line 19, with the following paragraph:

The boot process shown in FIG. 4A may be commenced by an initializing SML 50. Alternatively, the boot process may be directly invoked by a system administrator by, for example, a "boot" command. FIG. 5A is a screen shot showing an exemplary embodiment for providing such commands to the system administrator by the primary SML 50. In this embodiment, system administration commands are grouped as a set of "tabs" and displayed to the administrator. The administrator selects the tab containing the desired operations. FIG. 5A depicts an embodiment in which a "System Control" tab 54 provides four controls for a system: a "Power On" command 56 (depicted in gray to indicate the system is currently running); an explicit "Power Off" command 58; a "Reset" command 60; and a "System Interrupt" command 62. System information 64, as well as information 66 concerning the primary SML, [[66,]] is provided to the administrator. In the embodiment shown in FIG. 5A, the administration commands are provided using a browser-based user interface. Although FIG. 5A depicts an embodiment using NETSCAPE NAVIGATOR, manufactured by Netscape Communications of Mountain View, California, any browser may be used, including MICROSOFT INTERNET EXPLORER, manufactured by Microsoft Corporation of Redmond, Washington. A third way for the boot process shown in FIG. 4A to be invoked is by an SML following a system failure. This mechanism is discussed in greater detail below.

Please replace the paragraph beginning at page 11, line 16, with the following paragraph:

The boot process begins by determining a "boot list" (step 450[[)]] of FIG. 4A). A boot list is a list of component systems allowing the system to boot. For example, boot components may include processors, I/O controllers, BIOS, and other software (both application and system). In one particular embodiment, a boot list is an ordered list of processor-I/O controller pairs. In some embodiments, the boot list includes "heartbeat" values associated with each boot pair.

Heartbeat values are used by an SML 50 during system operation to determine if a processor 20 is functioning properly. Heartbeats are described in greater detail below. The boot list may be stored in a data structure that associates processor identification values with I/O controller values. For embodiments in which heartbeat values are also stored, the data structure includes an additional field to associate heartbeat timer values with each boot pair. The data structure may be stored on each SML 50 in a system 14'. In preferred embodiments, the data structure is stored in an non-volatile, erasable memory element, such as an EEPROM, that is accessible using auxiliary busses 60, 60'. In the event that the stored data structure is inconsistent (for example the data structure may include corrupted data values), or if the SML 50 is unable to retrieve data from the memory element (for example, if no memory element exists or if both auxiliary busses 60, 60' are not functioning), the SML 50 may use a hard-coded default list.

Please replace the paragraph beginning at page 13, line 1, with the following paragraph:

Once all system units are discovered by the SML 50, the SML 50 provides system clocks to the processors 20 and the I/O controllers 24 (step [[452]] 454). In other embodiments system clocks are not under the control of the SML 50 and, in these embodiments, step [[452]] 454 may be skipped.

Please replace the paragraph beginning at page 14, line 11, with the following paragraph:

Once the booting process is complete, or if the SML 50 determines that the system 14' should not be booted, the SML 50 enters a monitoring state (steps 412 or 464). In this state the SML 50 monitors heartbeat signals from each of the processors 20 to determine operation status of the system 14'. A failure to receive a heartbeat signal from a processor 20 during a predetermined period indicates that a failure has occurred. In this event, the SML 50 consults a non-volatile memory element to determine what actions, if ~~any~~ any, to take. The memory element may be the same memory element discussed above that stores the boot list, or a separate memory element may be provided that is accessible via the auxiliary busses 60, 60'. In one embodiment, the memory element stores a value that indicates one of seven actions for the SML 50 to take upon heartbeat failure: (1) no action; (2) normal interrupt; (3) non-maskable interrupt; (4) stop processor from executing; (5) system reboot; or (6) deterministic boot. Each of these options is discussed in detail below.

Please replace the paragraph beginning at page 16, line 20, with the following paragraph:

A memory value indicating "system reboot" allows the SML 50 to attempt to reboot the system in the event that ~~suspended~~ suspending a selected processor 20 does not succeed. The reboot process is similar to the reboot process described in connection with FIGs. 4 and 4A, except that the suspended processor 20 is skipped during reboot of the boot pairs listed in the boot list. To avoid repetitive heartbeat failure, the SML 50 maintains an index to identify the last processor-I/O boot pair in the boot list that last rebooted successfully. During the reboot process, this index is incremented to ensure that a different pair is selected as the starting pair each time. If successful, the state of the suspended processor 20 may be dumped for analysis, the state of the suspended processor 20 may be replaced with the state of one of the operational processors, or both. As above, if this mechanism doesn't succeed in restoring the system 14' to operational status, the SML 50 may dump the state of the suspended processor 20 for analysis by a system administrator, log the failure, alert an administrator to the failure, or any combination of these actions.

Please replace the paragraph beginning at page 17, line 12, with the following paragraph:

A memory value indicating "deterministic boot" allows the SML 50 to abandon the state of the suspended board and perform a full deterministic reboot, as described in connection with FIGs. 4 and 4A.

Please replace the paragraph beginning at page 18, line 10, with the following paragraph:

Each POP 110, 110' includes a POP server 114, 114' that is responsible for establishing and managing network connections to individual computer systems [[14,]] 14' and an address server 118, 118' that manages the assignment of IP addresses to computer systems 14'. In one embodiment, the address server 118[[']] is a Dynamic Host Configuration Protocol (DHCP) server. In another embodiment, the address server 118[[']] is a customized server application. In the embodiment shown in FIG. 6, computer systems [[14,]] 14' establish network connections with modem banks 116, 116' using a serial line protocol, such as the Point-to-Point (PPP) protocol or the Serial Line Internet Protocol (SLIP). The POP servers 114, 114' also establish packet routing and filtering functions to allow service personnel 182 connecting through the SPN

180 to access remote computer systems [[14,]] 14'. Although only two POPs 110, 110' are shown in FIG. 6, it should be understood that any number of POPs may be used to achieve geographic dispersity.

Please replace the paragraph beginning at page 19, line 10, with the following paragraph:

The remote access module 120 establishes and manages connections with computer systems [[14,]] 14'. The remote access server 120 may establish PPP connections for computer systems [[14,]] 14' either as an incoming call placed to the POP [[100]] 110 by the system 14' or as an outgoing call placed by the POP [[100]] 110 to the system 14'. In some embodiments, the remote access module 120 places a call to a system 14', authenticates itself to the system 14', and then terminates the call. In these embodiments, the system 14' places a return call to the POP [[100]] 110 to establish a connection. The POP [[100]] 110 may authenticate itself using predefined passwords, shared secrets, or public key infrastructure techniques.

Please replace the paragraph beginning at page 19, line 18, with the following paragraph:

The remote access module 120 communicates with an authentication server module 124 to authenticate systems 14'. The remote access module 120 monitors the state of all system connections and reports those changes to the connection server module 126. In certain embodiments, the remote access module 120 is provided as the RRAS portion of WINDOWS 2000, manufactured by Microsoft Corporation of Redmond, Washington. In other embodiments, the remote access module 120 is provided by a modified version of RRAS that supports the management of connections across multiple servers.

Please replace the paragraph beginning at page 20, line 4, with the following paragraph:

The authentication server module 124 verifies the authentication credentials of systems 14' and support personnel 182 seeking access to the POP [[100]] 110. In one embodiment, the authentication server module 124 verifies a username and password against a password database stored in the database 122. In other embodiments, the authentication server module 124 verifies an encryption key, digital certificate, or digital signature. In other embodiments, the authentication server module 124 includes accounting functionality that tracks accounting statistics relating to connections or connection attempts. In one embodiment, the authentication

server module is provided as the INTERNET AUTHENTICATION SERVICES module of WINDOWS 2000 manufactured by Microsoft Corporation of Redmond,[[.]] Washington. Once the system 14' or support personnel 182 is authenticated, the authentication server module 124 transmits a request for an IP address to the address server 118.

Please replace the paragraph beginning at page 20, line 15, with the following paragraph:

The database 122 stores information associated with connections. In some embodiments, the database 122 stores information associated with active connections, such as time of connection, frequency of connection requests, and addresses associated with particular requests. The database 122 can be provided as an ODBC-compliant, flat file, multidimensional, or relational database.

Please replace the paragraph beginning at page 21, line 6, with the following paragraph:

The connection server module 126 manages and directs the allocation of IP addresses to connections between the POP [[100]] 110 and the system 14'. The connection server module 126 is given an IP address by the address server 118, makes routing changes to assign that address to a connection, and transmits the address to SML 50 on the system 14'.

Please replace the paragraph beginning at page 21, line 10, with the following paragraph:

Connection requests from the centralized SPN 180 may originate directly from service personnel 182 or they may originate from the connection server module 126' of another POP server 114'. The SPN 180 and the various POPs [[100]] 110 may communicate using a variety of connections including standard telephone lines, LAN, or WAN links (e.g., T1, T3, 56kb, X.25), broad band connections (ISDN, Frame Relay, ATM) and wireless connections. Connections may be established using a variety of lower layer communication protocols (e.g. TCP/IP, IPX, SPX, NetBIOS, Ethernet, RS232, and direct asynchronous connection). In one embodiment, TCP/IP is used to communicate connection requests from the SPN 180 to the POP server 114.

Please replace the paragraph beginning at page 21, line 18, with the following paragraph:

Referring now to FIGs. 8, 8A, and 8B, the functional flow diagram depicts the operation of the described service network when allowing service personnel 182 connections to systems

14'. Service personnel 182 request connection to a system 14' (step 802). The service personnel 182 provide[[s]] an identifier of the system to which the connection is desired, as well as authentication credentials such as a user name and password or a digital certificate. The request is transmitted through the centralized SPN 180 to a POP [[100]] 110. The target POP [[100]] 110 may be predetermined, selected by the service personnel 182, or selected on the basis of information included in the identifier. For example, in some embodiments the centralized SPN 180 maintains a database of identifiers and associated POP addresses. When a request to connect to a particular site is received, the identification information is used to lookup the address of the POP [[100]] 110 with which the system 14' is associated. In certain embodiments, POPs [[100]] 110 are associated with certain geographical regions and are identified by IP addresses.

Please replace the paragraph beginning at page 22, line 9, with the following paragraph:

The connection server module 126 of the identified POP [[100]] 110 receives the connection request and validates the information associated with that request (step 820). If the authentication credentials associated with the request are not validated, the connection server module 126 denies access to the POP [[100]] 110 and returns a denial message to service personnel 182. If the authentication credentials associated with the request are valid, then the connection server module 126 registers the request (step 822). The request registration is stored in the database 122 and associated with an identifier. The identifier allows the connection request to be identified for use in subsequent communications. In some embodiments, other information is stored with the request such as the time and the system to which the request connection is made. The connection server module 126 returns a successful status message (step 824) to the service personnel 182.

Please replace the paragraph beginning at page 22, line 19, with the following paragraph:

The connection server module checks the database 122 to determine if the connection to the identified system 14' already exists (step 826). If a local connection already exists, then the connection server module 126 activates the connection, and selects one or more address filters (step 840), and the address filters are sent to the remote access module 120. In response to this message, the remote access module 120 sets the address filters (step 884). For example, in some instances the address filters are IP filters.

Please replace the paragraph beginning at page 23, line 4, with the following paragraph:

IP filters provide the client system 14' with security against SPN-side malicious activity, since the filters can be set to reject all packets except those from the SPN 180. If no local connection to the system 14' exists then the connection server module 126 broadcasts a message to all other POPs [[100]] 110 connected to the centralized SPN 180. The broadcast message polls the other connection server modules 126 to determine if they have an existing connection to the desired system 14'. The transmitted poll request includes the authentication credentials from the request.

Please replace the paragraph beginning at page 23, line 10, with the following paragraph:

Each of the other remote connection server modules 126' validates the poll request 870 and checks for a local connection by querying [[their]] its respective database[[s]] 122'. If no local connection exists, then the remote connection server module 126' does not respond to the broadcast message. Otherwise, the remote connection server module locks the connection to the system 14' (step 874) and sends a message to the connection server module 126 indicating that a local connection exists with the system 14' (step 876).

Please replace the paragraph beginning at page 23, line 16, with the following paragraph:

The connection server module 126 determines if a response has been transmitted to its polling requests (step 830). In some embodiments, the connection server module 126 waits a predetermined amount of time and if no response is received in that period of time, it is assumed that no response to the poll has been received. If no response is received, that indicates that no POP [[100]] 110 has a local connection to the desired system 14' and the connection server module 126 determines which connection server module 126 is the appropriate connection server module to initiate a local connection with the desired system 14'. In some embodiments, this determination can be based on geographical location, i.e., which connection server module 126 is the nearest to the desired system 14'. In other embodiments, this determination can be on the basis of the current processing activity in each POP [[100]] 110. If the connection server module 126 determines that it is the appropriate connection server module to initiate the local connection, then it initiates a connection with the desired system 14'.

Please replace the paragraph beginning at page 24, line 7, with the following paragraph:

If the connection server module 126 determines that it is not the appropriate connection server module to initiate the local connection, then connection server module 126 returns status to the service personnel 182 indicating that its request should be redirected to the identified connection server module 126' and the service personnel 182 transmits a connection request to the identified POP [[100]] 110 (step 802).

Please replace the paragraph beginning at page 24, line 12, with the following paragraph:

In some embodiments, when the connection server module 126 determines that it is not the appropriate connection server module to initiate the local connection, then the status message returned by the connection server module 126 causes the software used by service personnel 182 to automatically transmit a connection request to the identified POP [[100]] 110.

Please replace the paragraph beginning at page 24, line 16, with the following paragraph:

Referring [[back]] to step 880, the remote access module 120 initiates a connection with the desired system 14'. The system 14' requests authentication information (step 890) which is transmitted by the remote access module 120 (step 882). The system 14' authenticates the request and, if the authentication credentials are valid, allows access to the system 14'. In some embodiments, the system 14' terminates the serial connection (step 894) upon authentication and initiates a return serial connection based on the validated authentication credentials (step 896).

Please replace the paragraph beginning at page 25, line 1, with the following paragraph:

Once a system connection has been successfully established, the remote access module 120 requests an IP address from the authentication server module 124. The requested IP address is transmitted to the SML 50 on the client system 14'. In some embodiments, the IP address is transmitted using a remote procedure call. The assigned IP address allows communication with the system 14' to occur over the centralized SPN 180 and the POPs [[100]] 110 rather than the public Internet. In some embodiments, two IP addresses are assigned to a system 14'; one identifies the system 14'; and a second IP address identifies the SML 50.

Please replace the paragraph beginning at page 25, line 8, with the following paragraph:

Once a system connection has been successfully established, the remote access module 120 assigns an IP address to the SML 50 on the client system. The assigned IP address allows communication with the SML 50 over the centralized SPN 180 and the POPs [[100]] 110 rather than the public Internet. In some embodiments, two addresses are assigned: one to the SML 50 and one to the system 14'. In one embodiment, the IP address assigned to the system 14' is done through a remote procedure call.

Please replace the paragraph beginning at page 25, line 14, with the following paragraph:

The SML 50 uses the IP address transmitted to it by the remote access module 120 to control traffic at the client system 14'. IP filtering allows the SML 50 to block packets having associated addresses that are not intended for the system 14'.

Please replace the paragraph beginning at page 25, line 17, with the following paragraph:

In one detailed embodiment, the system 14' makes a connection to the POP/centralized SPN as follows:

Please replace the paragraph beginning at page 25, line 19, with the following paragraph:

1. If the system 14' is initiating the connection, it performs a remote procedure call ("RPC") to the SML 50 instructing it to establish a PPP connection to the POP/centralized SPN. The SML 50 can also initiate a connection for its own connection.

Please replace the paragraph beginning at page 26, line 2, with the following paragraph:

3. A POP/centralized SPN answers, the system 14' is authenticated and identified by the remote access module 120. A PPP session is established between the POP/centralized SPN and system 14'.

Please replace the paragraph beginning at page 26, line 7, with the following paragraph:

5. A POP/centralized SPN performs an RPC to the SML 50 to send a newly-assigned IP address T1 for the system 14'.

Please replace the paragraph beginning at page 26, line 13, with the following paragraph:

8. The system assigns this address to the system 14' side of the SML 50 virtual network interface.

Please replace the paragraph beginning at page 26, line 17, with the following paragraph:

10. The SML 50 takes note of the delivery IP address, and passes it onto the system 14' via a RPC.

Please replace the paragraph beginning at page 26, line 19, with the following paragraph:

11. The system takes note of the delivery IP address.

Please replace the paragraph beginning at page 27, line 6, with the following paragraph:

14. At this stage an IP connection now exists between the POP/centralized SPN and customer system 14'.

Please replace the paragraph beginning at page 27, line 16, with the following paragraph:

5. A POP/centralized SPN answers, the system 14' is authenticated and identified by the remote access module 120. A PPP session is established between the POP/centralized SPN and system 14'.

Please replace the paragraph beginning at page 28, line 1, with the following paragraph:

7. A POP/centralized SPN performs an RPC to the SML 50 to send a newly-assigned IP address T1 for the system 14'.

Please replace the paragraph beginning at page 28, line 7, with the following paragraph:

10. The system assigns this address to the system 14' side of the SML 50 virtual network interface.

Please replace the paragraph beginning at page 28, line 11, with the following paragraph:

12. The SML 50 takes note of the delivery IP address, and passes it onto the system 14' via a RPC.

Please replace the paragraph beginning at page 29, line 1, with the following paragraph:

17. The system 14' modifies its routing table to allow packets intended for the service system to be sent via the shared memory interface.

Please replace the paragraph beginning at page 29, line 8, with the following paragraph:

20. At this stage an IP connection now exists between the POP/centralized SPN and customer system 14'. Firewall functionality is implemented by the SML 50 rejecting any packet not addressed to T1 or T2, since only the customer system 14' and the POP/centralized SPN know addresses T1 and T2.

Please replace the paragraph beginning at page 29, line 12, with the following paragraph:

Additional steps are required to implement firewall functionality when the customer system 14' uses the Microsoft WINDOWS operating system. To communicate successfully through the firewall functionality, packets sent from the customer system 14' to the POP/centralized SPN must bear source address T1. If instead the packets bear the permanent address P1 of the customer system 14', then packets sent to the customer system 14' from the POP/centralized SPN will be rejected by the SML 50.

Please replace the paragraph beginning at page 30, line 4, with the following paragraph:

However, the desired value T2 cannot be used as the default gateway in the WINDOWS routing table because the SML 50 will not respond to Address Resolution Protocol (ARP) requests using the T2 address coming from the client system 14' side of the SML 50. The PPP interface bearing the T2 address is on the POP/centralized SPN side of the SML 50 and is not associated by the SML 50 with the client system 14' side of the SML 50. That is, the SML 50 is only responsive to ARP requests using the T2 address that come from the POP/centralized SPN side of the SML 50.

Please replace the paragraph beginning at page 30, line 14, with the following paragraph:

In one embodiment, this problem is solved by assigning temporary address T4 to the SML 50 side of the virtual network interface which, as discussed above, is also identified with address P2. The use of T4 as the default gateway lets WINDOWS set the source address of packets from

the client system 14' to T1 and, unlike the earlier scenario, the SML 50 will recognize and respond to ARP requests directed to the T4 address and coming from the client system 14' side of the SML 50.

Please replace the paragraph beginning at page 30, line 20, with the following paragraph:

Once a connection has been established with a client system 14', service personnel 182 can perform various operations on system 14' or access various parts of system 14' to monitor the system. Regardless of whether the SML 50 is in a boot or active state, it is in some embodiments useful for system personnel 182 to access video data corresponding to messages normally displayed on the display 26 of the system 14'. Such messages can provide valuable indicia of the state of the system 14' as well as each of its installed elements. For example, BIOS messages typically indicate the version of the BIOS that may or may not be compatible with the hardware version of the system 14'. BIOS messages can also indicate whether there is an incompatibility between the CPU versions in multiprocessor configurations that may affect the operations of the system 14'. Another type of fault indicia includes messages from I/O controllers 24 that indicate if the BIOS of the I/O controller 24 has been loaded and that also provide the status and configuration information for devices that it controls. Other types of fault indicia typically displayed on the display 26 of the system 14' include POST codes, memory contents, messages from software drivers, hardware and software interrupt messages, diagnostics results, etc.

Please replace the paragraph beginning at page 31, line 13, with the following paragraph:

In one embodiment and with reference to FIG. 9, the SML 50 comprises a PCI/PCI bridge 910, a VGA chip set 920 with associated VRAM 922, an arbiter 930, a PCI/Processor bridge 940, a processor 950, an inter-integrated circuits serial interface (I^2C) 952, a memory 954, and a network interface 956. The PCI/PCI bridge 910, such as a DEC 21153 PCI-PCI bridge/isolator, extends the system PCI bus 42 so that PCI devices on a local PCI bus 942 and sited on the SML 50 have visibility to the system 14'. An example of a PCI device that can be located on the SML 50 and which communicates via the local PCI bus 942 is the VGA chip set 920, such as the Cirrus Logic CL-GD5446 VGA controller. The VGA chip set 920 processes

and renders the video data stored in the VRAM [[622]] 922 for subsequent display on the server's display 26.

Please replace the paragraph beginning at page 32, line 1, with the following paragraph:

The PCI/Processor Bridge 940 (e.g., Tundra QSPAN PCI to Host bridge) enables the processor 950 (e.g., MPC860T I/O microprocessor and PowerPC core) to communicate with local and system PCI devices over a local processor bus 944 (e.g., Qbus). When performing a monitoring function, the processor 950 executes instructions stored in the memory 954 and accesses system and component information of the system 14' via I^2C logic 952 that has visibility on an I^2C bus, via the system PCI bus 42, and via the local PCI bus [[642]] 942. The processor 950 can also provide data to and receive instructions from a remote administrator via the network interface 956.

Please replace the paragraph beginning at page 33, line 10, with the following paragraph:

In one embodiment, referring to FIG. 10, the arbiter 930 includes two state machines: a PCI state machine 1000 that arbitrates access to the local bus 42 and a priority state machine 1002 that addresses blocking commands issued by the processor 950. The GRANT signal of the PCI state machine 1000 passes through the priority state machine 1002, which in turn decides whether the system 14' or the processor 950 has access to the VGA chip set 920.

Please replace the paragraph beginning at page 36, line 13, with the following paragraph:

If the priority of the system 14' is equal or the processor 950 has issued a blocking request and grant is asserted, then the priority state machine 1002 transitions from HOST1 state 1200 to LOCAL state 1204 through transition 1210. In LOCAL state 1204, the system 14' is blocked from accessing the bus 42. As long as the system 14' must be blocked, the priority state machine stays in LOCAL state through transition 1212. When blocking is no longer necessary, the finite state machine transitions to HOST1 mode through transition 1214 or HOST2 mode through transition 1216, depending on whether the system 14' has priority. Video may then be transmitted to service personnel 182 using an appropriate video transmission protocol, such as the Virtual Network Computing (VNC) protocol.